

Transformer モデルを用いた人流の時系列予測

挺屋 友幹¹ 片山 晋¹ 浦野 健太¹ 青木 俊介^{2,3} 米澤 拓郎¹ 河口 信夫^{1,2}

概要：人の流れをモデル化し、次の流れを予測する人流予測は渋滞予測や災害時の群衆誘導に活用されている。人流予測は時間的依存性を考慮するタスクと、空間的依存性を考慮するタスクの両方を考える必要がある。それらのタスクで最適化するために、空間依存性のモデル化には一般的に、畳み込みニューラルネットワーク（CNN）やグラフニューラルネットワーク（GNN）、グラフ畳み込みニューラルネットワーク（GCN）を用い、時間依存性モデル化には再帰的ニューラルネットワーク（RNN）ベースのモデルが用いられる。従来法ではそれら GCN と RNN を組み合わせたハイブリッドな深層学習モデルが提案されているが、RNN は時系列の逐次的な情報を捉えているに過ぎず、時系列で異なる時間ステップを等しく扱うために周期性を直接モデル化することができない。データを並列に扱う Transformer モデルも提案されているが、人流を予測するには不十分な点がある。そこで本研究では、GCN と Transformer を組み合わせた、人流の時系列予測手法を提案する。その提案モデルを実世界の人流データセットに対して適応させ、従来手法と予測精度を比較することで評価を行い、検証した評価指標において従来手法よりも高精度な予測が行えることが確認できた。

Time Series Prediction of Human Flow Using Transformer Model

Tomoki Choya¹ Shin Katayama¹ Kenta Urano¹ Shunsuke Aoki^{2,3} Takuro Yonezawa¹
Nobuo Kawaguchi^{1,2}

1. はじめに

人の流れをモデル化し、次の流れを予測する人流予測は渋滞予測や災害時の群衆誘導に活用されており、コンピュータ科学によって都市問題を解決するアーバンコンピューティングの問題の一つとして位置している。

人流における時系列予測を考える際、次の二つのタスクについて考える必要がある [1]。

(1) 変化する人流の時間的ダイナミクスを捉えるタスク

(2) 複雑な交通網における空間依存性を捉えるタスク

(1) の時間的依存性を捉えるタスクに関しては、電車の到着などのイベントの有無や天候の影響を受けて、非定常性を引き起こすことで長期予測を困難にしている。また、(2) の空間依存性を捉えるタスクに関しては、様々なスケー

ルの交通状況をモデル化するためには困難である [2][3]。例えばある通路での交通量は、下流が上流の交通量の影響を受ける。この影響の大きさは通路によって異なり、互いに影響し合う関係性をモデル化するのは難しい。それらの複雑な依存性を考慮して予測を行うのに、時系列に定常性があるものと仮定して行われる ARIMA モデル（自己回帰和分移動平均モデル）やベクトル自己回帰（VAR）などの単純な回帰モデルは有効ではなく、従来では深層学習を用いた手法がとられている。

深層学習手法としては (1) の時間依存性をモデル化するために、Gated Recurrent Unit (GRU) や Long Short Term Memory (LSTM) などの再帰的ニューラルネットワークが用いられている [4][5]。また、Li らは時間依存性を GRU でモデル化し、空間依存性をグラフ畳み込みニューラルネットワーク（GCN）を使用してモデル化する手法を提案した [1]。Lei らも同様に GCN と GRU を組み合わせ、GCN 内で計算に使われるグラフの隣接行列、つまりは空間的依存性を事前定義なしにモデル内で推論するハイ

¹ 名古屋大学大学院 工学研究科
Graduate School of Engineering, Nagoya University

² 名古屋大学未来社会創造機構
Institutes of Innovation for Future Society,
Nagoya University

³ 国立情報学研究所 National Institute of Informatics

ブリッドモデルを提案した [6].

しかしこれらの RNN ベースのモデルは時系列を逐次的にモデルに入力するため、時間的な逐次依存性を捉えることはできるが、時系列の異なるタイムステップを等しく扱うために時系列の周期性を直接モデル化することができない [7][8]. 人流には時間、日、週ごとなどに周期性が存在するのでこれは重要な欠点と言える。また、RNN ベースのモデルでは時系列を逐次的に入力するために、始めの方に入力した情報が潰れ、長期的な情報を保持しておくことが困難であるという欠点もある [9].

一方で 2017 年に Encoder-Decoder モデルである Transformer が提案された [10]. Transformer では attention 機構を多段に用いて入力のどこに注目して計算を行えば良いのかを学習する。したがって、入力に系列データを逐次的に入力する必要がなく、並列に処理を行うために RNN よりも計算効率が良い。入力データを並列に扱う際に Transformer では位置エンコーディングを適用することで入力の順序関係を埋め込んでいる。それらの機構を持つ Transformer はテキスト生成や機械翻訳などの分野で多く用いられている [11][12].

位置エンコーディングによって時系列の順序情報を付加できるため、Transformer を用いて時系列を予測する研究も行われている [13]. また、交通流予測にも Transformer が用いられており、GCN と Transformer を組み合わせたハイブリッドモデルが提案されている。[14][15][16]. しかし、これらのモデルでは GCN で用いるグラフの隣接行列にセンサ間の距離などに基づいた事前定義が必要であり、完全な空間的依存が考慮されないまま事前情報がバイアスとして働いてしまう。また、モデルの入力特徴量では、絶対時刻情報や付近のイベント情報などの、人流を予測するのに必要な特徴量が十分に考慮されていない。

本研究ではそれらの問題を解決するために、GCN と Transformer 内部に組み込み人流データの時間依存性と空間依存性をモデル化できる時系列予測手法を提案する。入力データには人流における時系列予測に関与する情報を結合し、人の通過情報だけでなくあらゆる情報を加味して予測を行えるようにする。また、人流には少なくとも 1 日の周期があることから、1 日前や 1 週間前などのデータには予測に大きく関わる局所的な時系列が存在する。その時系列を抽出し、Transformer の Decoder に使用する機構も提案する。その提案モデルを実世界の人流データセットに対して適応させ、従来の予測手法と予測精度を比較することで評価を行う。

本稿の構成は以下の通りである。2 章では関連研究について述べる。3 章では予測する対象となる環境について述べ、本研究における人流予測問題の定義を行う。4 章では提案手法について述べ、5 章では評価の内容や結果を議論する。最後に 6 章では本研究の結論と今後の課題について

述べる。

2. 関連研究

人流予測では深層学習を使ったモデルが多く提案され、モデルの多くは時系列を再帰的な処理によって計算し時系列の時間依存性を記憶する RNN が用いられてきた。通常の RNN は時系列の前方依存性のみを考慮しているが、Cui らによって後方依存性も捉えることのできる双方向 LSTM モデルが提案された [4].

しかし、それらの RNN ベースのモデルは時間依存性を捉える一方で、センサの設置位置などの地理的な情報を含む空間的依存性を考慮していない。Zhang らは時間依存性と空間的な近さの両方を考慮して予測するモデルを提案した [17]. そして同様に時間依存性と空間依存性の両方を捉えるために CNN や GNN を RNN ベースのモデルと組み合わせたハイブリッドモデルも提案された。Wu らは 1 次元畳み込みの CNN によって空間依存性を、LSTM によって時間依存性を学習するモデルで交通流予測を行なった [3]. また、Li らは CNN ではなくモデル内でグラフ構造を扱う GCN を GRU と組み合わせた Encoder-Decoder モデルである DCRNN を提案した。それまでの GCN ではあらかじめ定義された隣接行列が必要という問題を解決するために、モデル内で空間的依存性を表す隣接行列を学習する GCN と GRU を組み合わせたモデルの AGCRN も提案された [6]. しかしながら、時系列を逐次的に入力し再帰的な処理を行う RNN モデルでは時系列の逐次的な情報を捉えているに過ぎず、時系列で異なる時間ステップを等しく扱うために周期性を直接モデル化できないという欠点から予測精度に課題が生じている。また、処理の並列化が難しいため学習や予測の効率が悪く、長期記憶が難しいという欠点もある。

一方で Transformer が提案されてから、テキスト生成や機械翻訳で良い精度をあげ、モデルの汎用性の高さから最近では時系列予測にも用いられている [13]. 時間依存性と空間依存性の両方を捉える交通流予測タスクとしては、Xu らが位置埋め込みを学習可能なパラメータとして扱う GCN と Transformer を組み合わせたモデルを提案している [14]. また、Ling らも GCN と Transformer を組み合わせたモデルを提案し、位置エンコーディングの方法も提案したが、時間情報のみを埋め込み、その他の重要な情報 (例: 電車の時刻表、天気、付近のイベントの有無など) を考慮していない [16]. またそれらの GCN 内の隣接行列はセンサ間距離に基づいており、事前に定義しなければいけない。空間的依存性を考慮する際に、そのような事前定義情報だけでは完璧な空間情報を含めることはできず、逆にバイアスとして働いてしまう。

そこで我々は、時間以外の情報もモデルの入力に結合し、AGCRN にならって空間的依存性をモデル内で推論す

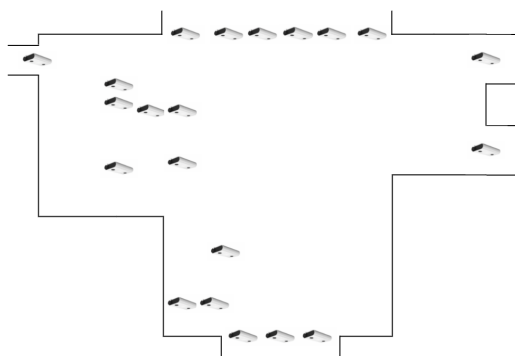


図 1 対象環境

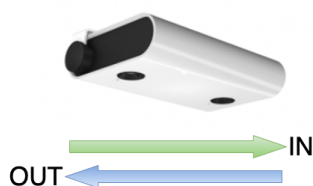


図 2 Vitracom(R) 社製 VC-3D センサ

る GCN-Transformer モデルを提案する。それに加え、予測に重要なであろう過去の時系列を抽出し、Decoder に入力する機構も提案する。本研究ではモデルの入力に電車の時刻表情報を使用するが、それが必要な対象として、人流の予測を行う環境について本稿の 3 章で述べる。

3. 人流予測問題の定義

本章では、本研究での人流予測問題の定義を行う。3.1 節では予測を行う対象の環境について述べ、3.2 節では予測問題の定義を行う。

3.1 対象環境

本研究で人流予測を行う環境は図 1 に示すような空間に人の通過を検知する 3D 通過センサを計 21 個設置した環境である。これは永田らによって行われた人流推定の研究 [18] と同様の環境となっている。使用された 3D 通過センサは、Vitracom(R) 社製の VC-3D (図 2) であり、図 1 に示された位置の天井に設置されていて空間に対して入った (IN)・出た (OUT) を計測する。本センサはプライバシー考慮のためにセンサ外部に画像を送信・保存しない設計となっており、人が通過した時刻や通過方向の他に身長やセンサの計測範囲を通過するのに要した時間も計測する。本研究ではこれらのうち、通過時刻と通過方向の情報を用いて人流予測を行う。また、この環境には図 1 における右側に電車のホームが隣接しており、電車の発着によって人の通行量が大きく変化する。

3.2 問題定義

行うタスクは過去の人流データ、つまりは過去のセンサ

が計測した人の通過情報を用いて将来の人の通過量を求めることである。通過人数を 15 分の時間粒度で集計し、センサあたり IN-OUT の 2 種類の通過方向の情報を持つことから、 21 (センサ数) $\times 2$ (IN-OUT) $= 42$ 個の時系列を持つ多変量時系列予測問題となる。また、空間的依存性をモデル化するために重みつき無向グラフ $G = (V, W)$ を考える。ここで、 V はセンサの IN-OUT が割り当てられ、 $|V| = 42$ であり、 W は時系列間の相関の強さを表す隣接行列である。ここで、時間長 M の X を入力し、将来の H の時間長を出力する問題は (1) 式のように定式化される。

$$X_{t+1}^{t+H} = F(G; X_{t-(M-1)}^t) \quad (1)$$

ここで F は入力を出力にマッピングする関数である。本研究ではこのような予測を行い、性能を評価する。

4. 提案手法

本章では提案手法の詳細について述べる。4.1 節では提案モデルの概要を説明し、4.2 節では Transformer を使って時系列予測を行う仕組みについて述べる。4.3 節では提案モデルで扱う GCN の構造について、4.4 節では提案手法で Transformer の外に設けたモジュールについて、そして 4.5 節では入力の特徴量について述べる。

4.1 提案モデルの構造

提案手法は Transformer をベースとした時系列予測モデルになっており、その構造の概要図を図 3 に示す。Transformer による時系列予測の仕組みについては 4.2 節で述べる。Transformer の Multi-head attention 機構には GCN が組み込まれており、Multi-head attention 機構内で空間依存性も考慮した計算を行う。GCN の定式化や Multi-head attention 機構の仕組みは 4.3 節で述べる。Transformer の外側には抽出モジュールがあり、ここでは Encoder の入力で使用する時系列よりも過去の時系列から、予測に大きく影響する局所的な時系列の抽出を行う。そして抽出モジュールによって抽出された時系列を Decoder の入力の 1 つとして用いる。抽出モジュールの詳細については 4.4 節で述べる。また、モデルへの入力には通過した人数の情報だけでなく、絶対時刻情報や電車の発着時刻の情報も結合させる。その入力特徴量については 4.5 節で述べる。

4.2 Transformer による時系列予測

Transformer は attention 機構を用いた Encoder-Decoder モデルであるが、Encoder では過去時系列を Positional Encoding を適用した後に attention 機構を用いて時系列情報情報を符号化する。Encoder からの出力は Decoder の Multi-head attention 機構に入力され、符号化された入力情報をもとに Decoder で予測が行われる。Encoder には予測時点 t までの任意の時間長 M の $X_{t-M}, X_{t-M+1}, \dots, X_t$

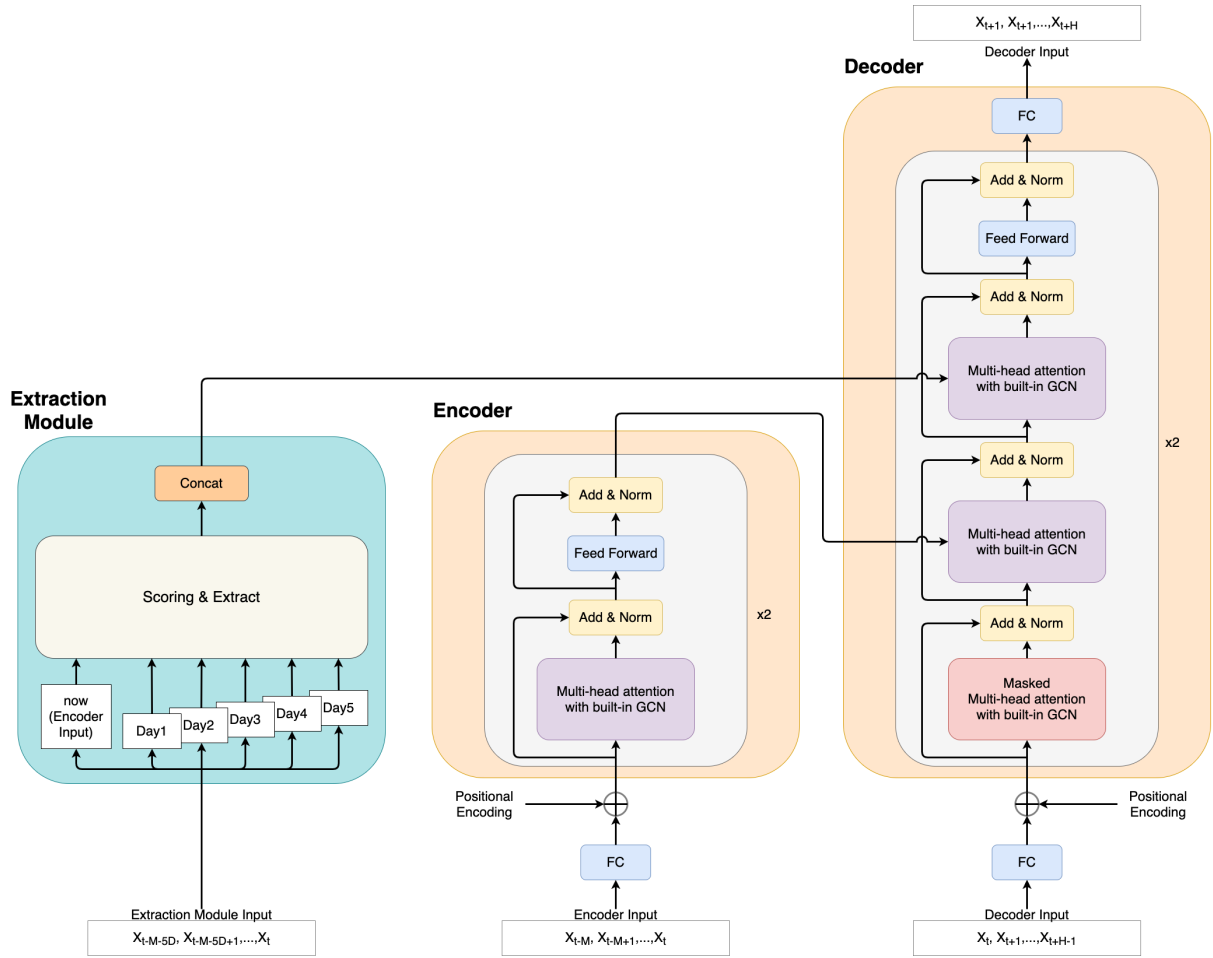


図 3 提案手法の構造

を入力し，Decoder には予測時点 t から $t+H-1$ までの $X_t, X_{t+1}, \dots, X_{t+H-1}$ を入力する．ここで H は予測時系列長であり， H 点先の予測までを行うことを意味する．そして Decoder から時刻 $t+1$ から $t+H$ までの予測値 $X_{t+1}, X_{t+2}, \dots, X_{t+H}$ が出力される．Decoder の入力にはすでに予測先の情報が含まれているが，Decoder 内の Masked Multi-head attention 機構では将来の値を参照して予測を行わないように mask 処理を行って学習する．推論フェーズでは時刻 t から順に Decoder に入力し，予測した値を時刻 $t+1$ の入力として使用する．

4.3 GCN の組み込み

本節では，GCN を Transformer の Multi-head attention 機構に組み込む機構の定式化を行う．

人流の予測には時系列の空間的依存性を捉えるために GCN を導入する．GCN で行われるグラフ畳み込み演算は，1 次のチェビシェフ多項式展開によって近似でき，(2) 式のように一般化される．

$$\mathbf{Z} = \left(\mathbf{I}_N + \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}} \right) \mathbf{X} \Theta + \mathbf{b} \quad (2)$$

ここで， $\mathbf{A}$ はグラフの隣接行列であり $\mathbf{A} \in \mathbb{R}^{N \times N}$ であり，

N はグラフのノードの数，つまりは時系列の数で本研究では 3.2 節で述べたように $N = 42$ である． $\mathbf{D}$ は次数行列， $\mathbf{X} \in \mathbb{R}^{N \times C}$ および $\mathbf{Z} \in \mathbb{R}^{N \times F}$ は GCN 層への入出力， $\Theta \in \mathbb{R}^{C \times F}$ や $\mathbf{b} \in \mathbb{R}^F$ は重みとバイアスのパラメータであり，学習の過程で最適化される．ただし C はチェビシェフ多項式展開の次数， F は任意の出力次元である．このグラフ畳み込みでは重みプールやバイアスプールが全てのノードで共有されることになり，個々の時系列の特徴を学習することなく，全ての時系列に共通する特徴のみ学習することとなる．日中に通行量が増え，夜中には減るなどの共通した特徴は学習できるが，人流予測では人の移動の特徴を捉えるためにセンサそれぞれの細かな特徴を捉える必要がある．そこで，本研究では AGCRN モデル [6] に基づいて，グラフ畳み込み演算を (3) 式で表す．

$$\mathbf{Z} = \left(\mathbf{I}_N + \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}} \right) \mathbf{X} \mathbf{E}_G \mathbf{W}_G + \mathbf{E}_G \mathbf{b}_G \quad (3)$$

(2) 式と (3) 式では重みやバイアスのパラメータに $\mathbf{E}_G \in \mathbb{R}^{N \times d}$ をかけているところが異なる． d はノード埋め込み次元を表し，重みプールは $\mathbf{W}_G \in \mathbb{R}^{d \times C \times F}$ として表され， Θ は $\Theta = \mathbf{E}_G \cdot \mathbf{W}_G$ の演算を経て得られる．このような書き換えによって，ノード固有の重みプールやバイアスプー

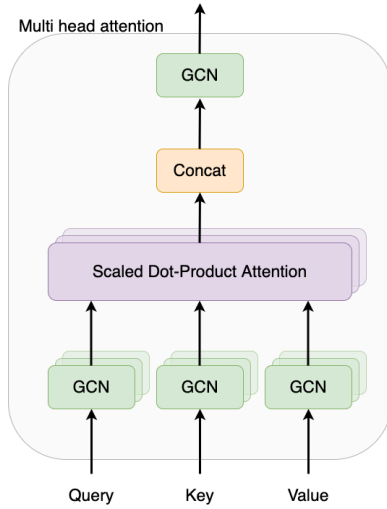


図 4 GCN の attention への組み込み

ルが確保でき、ノード固有の特徴、つまり人流を計測したセンサ個々の時系列の特徴を学習できる。

また、(2) 式では予め定義された隣接行列 A が用いられているが、事前にグラフを定義すると空間依存性に関する全ての情報を含めさせることは難しく、逆に隣接行列に設定した値がバイアスとしてはたらいってしまうことが懸念される。そこで A を生成して対称正規化ラプラシアン行列 $D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$ を得るのではなく、ノード埋め込みベクトルを使用した低ランク近似を行い、対称正規化ラプラシアン行列を次のように求められる。

$$D^{-\frac{1}{2}}AD^{-\frac{1}{2}} = \text{softmax}(\text{ReLU}(\mathbf{E}_G \cdot \mathbf{E}_G^T)) \quad (4)$$

これらの GCN の演算過程を Transformer の Multi-head attention 機構の線形回帰部分と置き換える ((5) 式)。この構造をとることで空間依存性を考慮した attention を求めることができる。GCN を Multi-head attention 機構に組み込んだ概要図を図 4 に示す。

$$\begin{aligned} \tilde{\mathbf{A}} &= \text{softmax}(\text{ReLU}(\mathbf{E}_G \cdot \mathbf{E}_G^T)) \\ \mathbf{Q} &= \tilde{\mathbf{A}}\mathbf{X}\mathbf{E}_G\mathbf{W}_Q + \mathbf{E}_G\mathbf{b}_Q \\ \mathbf{K} &= \tilde{\mathbf{A}}\mathbf{X}\mathbf{E}_G\mathbf{W}_K + \mathbf{E}_G\mathbf{b}_K \\ \mathbf{V} &= \tilde{\mathbf{A}}\mathbf{X}\mathbf{E}_G\mathbf{W}_V + \mathbf{E}_G\mathbf{b}_V \\ \text{Attention} &= \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V} \end{aligned} \quad (5)$$

4.4 抽出モジュール

本節では、予測時点付近のデータだけでなく、広く過去の時系列を入力に使用する際に行う操作について述べる。

人は 1 日単位で行動するため、人流データには少なくとも 1 日の周期が存在する。そのため予測したい対象時系列と似た時系列は 1 日前や 1 週間前にも見られ、その付近のデータは予測に有効である [19]。1 週間分のデータをその

まま深層学習モデルに入力しても予測はある程度行えるが、入力が非常に大きくなるため計算時間が大幅に増大し、非常に多くの計算リソースが必要となる。

そこで本研究では 1 日前や 1 週間前などの過去系列から、予測時点付近と似た動きをしている局所的な時系列を探索し、抽出するモジュールを使用する。

図 3 の Extraction Module がそれにあたり、このモジュールには Transformer の Encoder に入力する時系列に加えてそれよりも過去数日間のデータを投入する。抽出モジュール内部ではまず日毎にデータを分割してスコアリングし、抽出機構に入力する。ここでは、予測時点までの数時間分と似た時系列を過去数日分のデータそれぞれから探索する。時系列の類似性をスコアする方法はユークリッド距離とコサイン類似度を用いて (6) 式のように求める。

$$\begin{aligned} d(x, y) &= \sqrt{(x_1 - y_1)^2 + \dots + (x_n - y_n)^2} \\ c(x, y) &= 1 - \frac{x_1y_1 + \dots + x_ny_n}{\sqrt{x_1^2 + \dots + x_n^2}\sqrt{y_1^2 + \dots + y_n^2}} \quad (6) \\ val &= d(x, y) * c(x, y) \end{aligned}$$

ここで $x_1, x_2, \dots, x_n$ は系列 X の要素で、 $y_1, y_2, \dots, y_n$ は系列 Y の要素である。求められたスコアのうち最も小さい値の時系列が予測時点と似た時系列とみなされ、その次の時系列が予測したい時系列と似た時系列になるとして抽出される。抽出された日毎の時系列は最後に結合され、その後 Decoder の入力の 1 つとして Multi-head attention 機構に入力される。人が直感的に時系列予測を行う場合、過去の同じような動きをしているところを参考に予測を行うだろう。例えば、昨日と一昨日のこの時間に急激に値が増加したので、今日のこの時間もおそらく増加するだろうといった考え方である。抽出モジュールの出力を Decoder で用いることで深層学習的予測と人の感覚に近い予測の両方の側面をもって予測を行うことができる。

4.5 入力特徴量

RNN へは逐次的に時系列を入力するため、時系列の順序関係は担保されている。一方で Transformer は入力時系列を並列に扱うため時系列の順序関係が保証されていない。そこで Transformer では Positional Encoding を行うことで時系列の相対的な位置についての情報を付加する。人流には、朝 8 時頃に通勤ラッシュで人が多くなるなどの絶対時間が関わってくる。そこで本研究では通常の Positional Encoding だけでなく、時刻・曜日・電車の発着情報・時間遅延埋め込みを時系列に結合する。操作は次の通りである。

- (1) 時刻・曜日・電車の発着情報・時間遅延埋め込みを時系列に結合する
- (2) 全結合層 (FC) に通す
- (3) 通常の Transformer で使われている Positional Encoding を行う

Multi-head attention 機構では入力の特徴量方向に任意の数の分割して多重の attention を行う。その際に全結合層に通さないと head によって見られる情報に差が生じ、全ての特徴量にまたがって attention 機構を用いることができない。そこで (2) の操作として全結合層に通すことで単一の head で全ての特徴量を加味できるようにした。以下で (1) で用いた特徴量と (3) の Positional Encoding について述べる。

時刻

時刻情報は月、日、時、分のそれぞれを (7) 式のように sin と cos を用いて連続性のある形に変換し、計 8 次元を結合した。t はその時刻におけるそれぞれの値、T は各周期であり $T_{\text{月}} = 12$, $T_{\text{日}} = 31$, $T_{\text{時}} = 24$, $T_{\text{分}} = 60$ とした。

$$\begin{aligned} \text{TimeEmbedding}_{\sin} &= \sin\left(\frac{2\pi}{T}t\right) \\ \text{TimeEmbedding}_{\cos} &= \cos\left(\frac{2\pi}{T}t\right) \end{aligned} \quad (7)$$

曜日

曜日情報は 7 次元の one-hot ベクトル表現とした。月曜の場合は $[1, 0, 0, 0, 0, 0, 0]$ となる。

電車の発着情報

電車の情報には、電車の種類 (普通・急行など) が 6 種類と車両数がある。それらと発着の 2 種類を組み合わせた 12 次元のベクトルとして表現した。ここでは $2(\text{発着}) \times 6(\text{種類}) = 12$ の one-hot ベクトルに対し、車両数をかける操作を行なった。例えばある時間に 8 両編成の普通列車が到着する場合は、 $[8, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]$ となる。

時間遅延埋め込み

RNN モデルでは入力の情報を記憶する機構があり、各時間の隠れ表現には前にどのような入力があったかの情報が保持されている。Transformer にはそのような機構はないが、入力の特徴量として前の時系列情報を連結させることで時系列の情報の蓄積を模倣するという時間遅延埋め込みを本手法では用いる。時間遅延埋め込みは (8) 式で表される。本手法では 5.1 節で述べる予測時系列長と同じ $d=8$ とし、前の 8 個の時系列を結合した。

$$TDE_{d,\tau}(x_t) = (x_t, x_{t-\tau}, \dots, x_{t-(d-1)\tau}) \quad (8)$$

Positional Encoding

Transformer では入力の相対的な位置関係を保証するために (9) 式のように d -次元空間でトークンを生成し、そのトークンを各入力に足し合わせる。本手法では入力の全結合層の直後で Positional Encoding を行なった。

$$\begin{aligned} PE_{(pos,2i)} &= \sin\left(pos/10000^{2i/d_{\text{model}}}\right) \\ PE_{(pos,2i+1)} &= \cos\left(pos/10000^{2i/d_{\text{model}}}\right) \end{aligned} \quad (9)$$

ここで d_{model} は特徴量の次元であり、通行量 (1) + 時刻

表 1 深層学習モデルのパラメータ

パラメータ	値
Encoder への入力長	100
Decoder への入力長	8
Decoder の出力長	8
multi head 数	8
epoch 数	1000
バッチサイズ	32
ノード埋め込み次元	20
最適化法	Adam
Adam beta1	0.9
Adam beta2	0.98
損失関数	MAE
チェビシェフ次数	2
データの正規化	std
Dropout	0.1
ペナルティ項の係数	0.001

(8) + 曜日 (7) + 時刻表 (12) + 時間遅延埋め込み (8) = 36 次元である。

5. 評価

本章ではモデルの評価の詳細や結果について述べる。

5.1 データセット

使用するデータセットは、3.1 節で述べた環境で集計された人流データセットである。3D 人通過センサから収集された人の通過データを、15 分の時間粒度で通過人数をカウントしたものである。データの期間は 2020 年 6 月 1 日から 2020 年 11 月 18 日の期間の平日であり、不慮の原因でデータが収集されていない部分は事前に取り除いた。このデータのうち、はじからの 6 割を訓練、その後の 2 割を検証、最後の 2 割をテストに用いた。なお訓練時には訓練データを検証データをシャッフルさせた。

5.2 評価指標

予測精度の評価には MAE(Mean Absolute Error, 平均絶対誤差), RMSE(Root Mean Square Error, 平均平方二乗誤差), RRSE(Root Relative Squared Error, 相対平方二乗誤差), CORR(相関係数, Correlation Coefficient) を用いる。MAE, RMSE, RRSE では評価値が小さいと良い精度で予測できていることを表す。CORR では評価値が 1 に近いと良い精度で予測できていることを表す。

5.3 実験内容

5.1 節のデータセットを学習させて予測を行い、5.2 の評価指標を用いて提案手法と従来法の予測精度比較を行う。1 日での周期は存在することから、それ以上の時間長である 25 時間分のデータを入力し、2 時間先までの予測を行う予測タスクを行う。比較対象の従来法としては LSTM

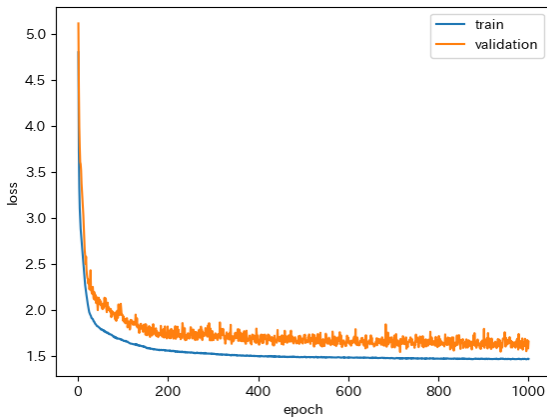


図 5 訓練と検証での損失の推移

表 2 モデルの評価比較

	評価指標			
	MAE	RMSE	RRSE	CORR
LSTM	6.21	17.35	0.90	0.48
ARCRN	5.86	16.82	0.94	0.52
SimpleTransformer	4.62	13.22	0.61	0.58
提案手法	4.53	12.12	0.54	0.58

と ARCRN と単純な Transformer を用いた。提案手法の Transformer モデルと区別するために 5.4 節の結果ではこれを SimpleTransformer とした。

モデルの学習には表 1 で示すパラメータを使用し、学習の安定を図るために損失関数には学習パラメータの L2 正則化をペナルティ項として付加した ((10) 式)。

$$\mathcal{L}(\mathbf{W}_\theta) = \sum_{i=t+1}^{i=t+\tau} |\mathbf{x}_{:,i} - \mathbf{x}'_{:,i}| + \lambda \sum_i |\boldsymbol{\Theta}_i|^2 \quad (10)$$

提案手法の学習過程での損失の推移を図 5 に示す。図 5 は縦軸が損失の大きさと横軸がエポック数を表し、青の線が訓練の損失を、オレンジの線が検証の損失を表す。検証の損失が最も低いモデルを保存しテストに使用した。

また、実装には深層学習ライブラリ Pytorch を用い、バージョンは 1.8.0 である。実行は NVIDIA A100 Tensor core GPU で行った。

5.4 結果

実験の結果を表 2 に示す。4 つの評価指標全てにおいて提案手法が良い精度であることがわかる。従来法の中では SimpleTransformer が全ての指標において良い精度を出しているが、それに対して MAE では 4.62 から 2 %, RMSE では 13.22 から 8.3 %, RRSE では 0.61 から 11.5 % の精度向上が見られた。CORR に関しては SimpleTransformer と同等の精度となった。

また、各手法のテストデータでの予測の様子を図 6 に示

す。図 6 は通行量の多いあるセンサでの IN 方向と OUT 方向の図であり、縦軸が通過人数、横軸が時刻となっている。青の線が正解データ、オレンジの線が LSTM による予測値、緑の線が AGCRN による予測値、赤の線が SimpleTransformer による予測値、紫の線が提案手法による予測値を表す。全体的に見て提案手法がよりフィットしていて正確に予測できていることが伺える。このセンサのもとでは人の通行量は細かく大きく変化するために予測が難しい。その環境でも提案モデルは他のモデルと比べて正確に急激な通行量の増加にも対応している。特に OUT 方向の時刻 22 時付近での大きく立ち上がっている部分では違いが歴然であり、提案モデルのみが通行量が大きく増加することを予測している。LSTM は提案手法と比べて時系列の立ち上がりを予測できていない箇所や、間違って立ち上がりを予測している部分が多く見られる。LSTM のような時間方向のみを見るモデルと比べて、より正確な時間依存性の捕捉によって間違った時系列の増減が少なくなり、そして GCN による空間依存性の捕捉によってより正確に通行量の増減を予測できるようになったと考えられる。SimpleTransformer と比べてもよく予測できている箇所が多いことから、GCN の組み込みや抽出モジュールが良く予測結果に寄与したと考えられる。しかし一方で通行量の増加に対応できてない箇所も多く存在する。間違った予測をしている部分への解釈を行うことが今後の課題となる。

6. まとめ

本稿では Transformer モデルを用いた人流の時系列予測手法の提案を行なった。人流予測での時間的依存性や空間的依存性の考慮に対して、従来では空間依存性のモデル化に CNN や GNN, GCN を用いられ、時間依存性モデル化には LSTM などの RNN ベースのモデルが用いられているが、RNN では時系列の逐次的な情報を捉えているに過ぎず、時系列で異なる時間ステップを等しく扱うために周期性を直接モデル化できないという問題がある。さらには、RNN は入力に対して逐次的な操作を行うために並列化が難しく、学習や予測の効率が悪く、長期記憶が難しいという問題もある。また、既存の Transformer と GCN を組み合わせたモデルでは隣接行列の事前定義が必要で、入力特徴量が人流を予測するには不十分であった。そこで事前定義が不要な GCN と Transformer を組み合わせ、人流データの空間依存性と時間依存性をモデル化する時系列予測手法を提案した。提案モデルでは予測に関連性の高い時系列を抽出するモジュールも用い、入力特徴量には様々な情報を結合させ、その後相対位置を付加するハイブリッドな位置埋め込み手法を用いた。提案モデルを実世界の人流データセットに対して適応させ、既存手法と予測精度を比較して評価を行い、検証した評価指標において従来手法よりも高精度に予測できることが確認できた。予測した結果

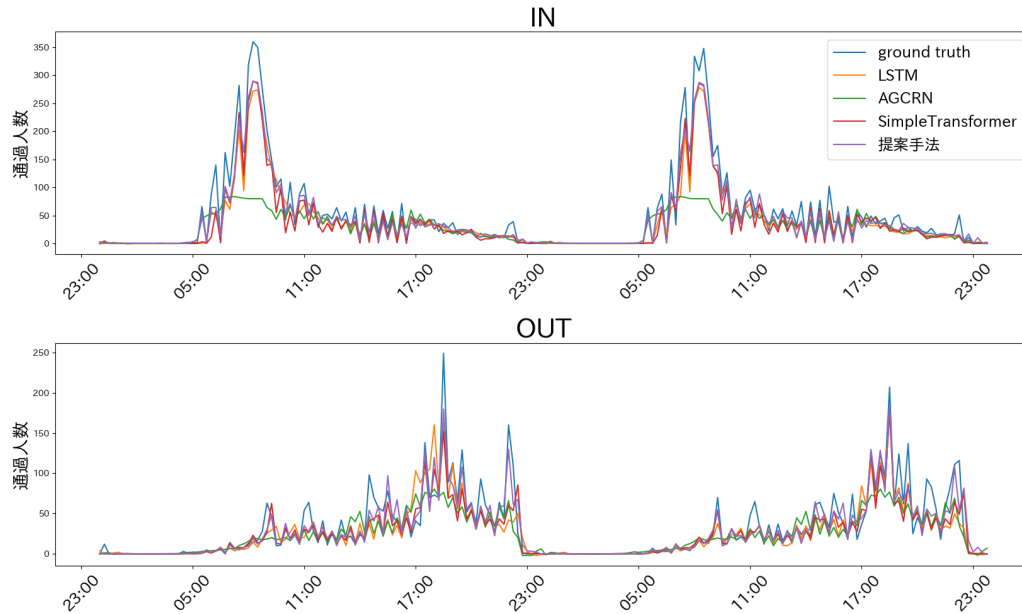


図 6 あるセンサでの予測結果

をもとにモデルへの解釈性を高め、今後は更なる精度向上を図る。

謝辞 本研究の一部は、NICT 委託研究 (21406)、総務省委託研究 SCOPE (0159-0110)、JST COI-NEXT (JP-MJPF2006) の支援を受けています。

参考文献

- [1] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data driven traffic forecasting. In the Sixth International Conference on Learning Representations (ICLR), 2018.
- [2] Davis N, Raina G, and Jagannathan K. Grids versus graphs: Partitioning space for improved taxi demand - supply forecasts. Preprint, arXiv:1902.06515, 2020.
- [3] Wu Y, Tan H, Qin L, Ran B, and Jiang Z. A hybrid deep learning based traffic flow prediction method and its understanding. Transportation Research Part C: Emerging Technologies, 90, 166–180, 2018.
- [4] Cui Z, Ke R, and Wang Y. Deep bidirectional and unidirectional LSTM recurrent neural network for network - wide traffic speed prediction. International Workshop on Urban Computing (UrbComp), 2018.
- [5] Jin W, Lin Y, Wu Z, and Wan H. Spatio - temporal recurrent convolutional networks for citywide short - term crowd flows prediction. In Proceedings of the Second International Conference on Compute and Data Analysis, 2018.
- [6] Lei Bai, Lina Yao, Can Li, Xianzhi Wang, Can Wang. Adaptive Graph Convolutional Recurrent Network for Traffic Forecasting, arXiv preprint arXiv:2007.02842, 2020.
- [7] Guo S, Lin Y, Feng N, Song C, and Wan H. Attention based spatial - temporal graph convolutional networks for traffic flow forecasting. In Proceedings of the 33rd AAAI Conference on Artificial Intelligence, 2019.
- [8] Yao H, Tang X, Wei H, Zheng G, and Li Z. Revisiting spatial - temporal similarity: A deep learning framework for traffic prediction. In Proceedings of the 33rd AAAI Conference on Artificial Intelligence, 2019.
- [9] Khandelwal U, He H, Qi P, and Jurafsky D. Sharp nearby, fuzzy far away: How neural language models use context. In Proceedings of the 56th Meeting of the Association for Computational Linguistics, 2018.
- [10] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez A N, Kaiser L, Polosukhin I. Attention is all you need. In Proceedings of the 31st Conference on Neural Information Processing Systems, 2017.
- [11] Dai Z, Yang Z, Yang Y, Carbonell J, Le Q V, and Salakhutdinov R. Transformer - XL: Attentive language models beyond a fixed - length context. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (pp. 2978–2988), 2019.
- [12] Devlin J, Chang M W, Lee K, and Toutanova K. BERT: Pre-training of deep bidirectional Transformers for language understanding. Preprint, arXiv:1810.04805, 2018.
- [13] Li S, Jin X, Xuan Y, Zhou X, Chen W, Wang Y X, and Yan X. Enhancing the locality and breaking the memory bottleneck of Transformer on time series forecasting. In Proceedings of the 32nd Conference on Neural Information Processing Systems, 2019.
- [14] Xu M, Dai W, Liu C, Gao X, Lin W, Qi G J, and Xiong H. Spatial - temporal Transformer networks for traffic flow forecasting. arXiv:2001.02908, 2020.
- [15] Li Y, and Moura J M. Forecaster: A graph Transformer for forecasting spatial and time dependent data. arXiv:1909.04019, 2019.
- [16] Ling Cai, Krzysztof Janowicz, Gengchen Mai, Bo Yan, Rui Zhu, Traffic transformer: Capturing the continuity and periodicity of time series for traffic forecasting. Transactions in GIS, 2020.
- [17] Zhang J, Zheng Y, Qi D, Li R, and Yi X. DNN - based prediction model for spatio - temporal data. In Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, 2016.
- [18] Yoshiteru Nagata, Takuro Yonezawa, and Nobuo Kawaguchi. Person-Flow Estimation with Preserving

Privacy using Multiple 3D People Counters. Science and Technologies for Smart Cities 2021 (Transaction of 5th EAI International Conference on IoT in Urban Space), 2020.

- [19] 挺屋 友幹, 青木 俊介, 米澤 拓郎, 河口 信夫. 時間・空間的相関を考慮した人流の時系列予測手法. 情報処理学会研究報告ユビキタスコンピューティングシステム (UBI), 2020.